

# The challenges of unbounded treewidth in parameterised subgraph counting problems \*

Kitty Meeks

School of Mathematical Sciences, Queen Mary University of London

`k.meeks@qmul.ac.uk`

February 2014

## Abstract

Many of the existing tractability results for parameterised problems which involve finding or counting subgraphs with particular properties rely on bounding the treewidth of the minimal subgraphs having the desired property. In this paper, we give a number of hardness results – for decision, approximate counting and exact counting – in the case that this condition on the minimal subgraphs having the desired property does *not* hold. These results demonstrate that in some cases the bounded treewidth condition is necessary for the existence of an efficient algorithm, and lead to two dichotomies for problems which involve finding or counting multicolour subgraphs.

## 1 Introduction

The graph parameter treewidth plays a crucial role in many results relating to parameterised counting complexity, a field introduced by Flum and Grohe in [10]. This is true in particular for subgraph counting problems, informally those problems which can be phrased as follows:

---

\*Research supported by EPSRC grant “Computational Counting”

*Input:* An  $n$ -vertex graph  $G = (V, E)$ , and  $k \in \mathbb{N}$ .

*Parameter:*  $k$ .

*Question:* How many (labelled)  $k$ -vertex subsets of  $V$  induce graphs with a given property?

It should be noted that, although we refer to induced subgraphs, problems in which the goal is to count non-induced copies of a given graph  $H$  can just as well be formulated in this manner: it suffices to define the property we are looking for to be “contains  $H$  as a subgraph”.

A formal model for subgraph counting problems of this kind was introduced in [13], and this is the model we use here; in Section 2 we give the full definition of this model and its multicolour version (introduced in [14]), as well as defining natural extensions of these two models to formalise the corresponding decision problems.

Many problems falling within the scope of this model have received attention in the past, including  $p$ -#CLIQUE,  $p$ -#PATH and  $p$ -#CYCLE [10],  $p$ -#MATCHING [6],  $p$ -#INDUCED SUBGRAPH ISOMORPHISM( $\mathcal{C}$ ) [5], and  $p$ -#CONNECTED INDUCED SUBGRAPH [13], all of which are known to be intractable from the point of view of parameterised complexity. While there are no non-trivial problems in this family which are known to be tractable from the point of view of exact counting in the uncoloured setting, there are nevertheless a number of positive results regarding approximate counting, many of which exploit the treewidth of the subgraphs to be counted. Arvind and Raman [3] demonstrated that there is an efficient parameterised approximation algorithm to compute the number of (not necessarily induced) copies in a graph  $G$  of any given  $k$ -vertex graph  $H$  of bounded treewidth, while Jerrum and Meeks [13] generalised this to show that the same is true for any monotone property provided that all minimal labelled subgraphs with the desired property have bounded treewidth.

In this paper, we are concerned with the situation in which this last condition does not hold; we show that in a number of situations, if it is not true that all minimal unlabelled subgraphs satisfying a (not necessarily monotone) property have bounded treewidth, the problems considered become intractable (up to standard assumptions in parameterised complexity). In particular, in this setting we prove

- exact counting is intractable from the point of view of parameterised complexity, and

- in the related problem in which subgraphs must also be *colourful* (with respect to a colouring of the graph specified as part of the input), both the decision version and approximate counting are intractable from the point of view of parameterised complexity.

Notions of parameterised intractability and approximability will be made precise in Section 1.2, and some examples of problems to which these results can be applied will be given in Section 2.3.

Some sufficient conditions for an exact counting problem in this model to be  $\#W[1]$ -complete were established in [14]; the new hardness results here provide a different kind of sufficient condition for intractability, and so substantially increase the restrictions on what problems in the model could possibly be in FPT from the point of view of exact counting. Additionally, our results provide another source of examples of  $\#W[1]$ -complete parameterised counting problems, of which relatively few are known at present.

Combined with existing tractability results, our hardness results give rise to dichotomies for the parameterised complexity of *multicolour* subgraph counting problems from the point of view of decision and approximate counting, when the property in question is monotone, and with the additional restriction in the case of approximate counting that the minimal labelled subgraphs having the property are also minimal when regarded as unlabelled subgraphs. Moreover, it turns out that the conditions required for tractability of both decision and approximation in this case are exactly the same, so in this situation an efficient algorithm for the decision problem implies that there exists an efficient approximation algorithm for the counting version. These results concerning the approximability of the multicolour version make some progress towards a characterisation of which problems in this model admit an FPTRAS, although there is much more work to do in order to provide a good answer to this question. The dichotomies have a flavour reminiscent of that proved by Grohe [12] concerning the complexity of deciding whether there exists a homomorphism from a fixed graph  $H$  to the input graph  $G$ , and also the result due to Dalmau and Jonsson [7] for the counting version of this problem: in this different setting, the decision problem is known to be tractable if and only if the core of  $H$  has bounded treewidth, while the counting version is tractable if and only if  $H$  has bounded treewidth itself.

The proofs of our hardness results are based on a construction that exploits the characterisation of graphs of unbounded treewidth given by the Excluded Grid Theorem of Robertson and Seymour [16]. This deep combi-

natorial result was previously used by Grohe [12] in proving the hardness of the graph homomorphism problem mentioned above.

The remainder of the paper is organised as follows. In the rest of this section, we introduce the main notations and definitions that will be used throughout the paper, and mention some key background from the theory of parameterised counting complexity. In Section 2, we provide more detailed background, first describing the model we use for subgraph counting problems and some known facts about it, and giving an overview of existing results concerning the complexities of problems within the model, then introducing formally the notions of graph minors and treewidth, which are crucial to our later proofs, before giving some examples of problems in this model to which our results apply. The main technical content is in Section 3, where we describe the construction we will use for all our complexity results, and demonstrate its important features. Finally, all of our complexity results are proved in Section 4.

## 1.1 Notation and definitions

Throughout this paper, all graphs are assumed to be simple, that is they do not have multiple edges or self-loops. The order of a graph  $G$ , written  $|G|$ , is the number of vertices in  $G$ . Given a graph  $G = (V, E)$ , and a subset  $U \subset V$ , we write  $G[U]$  for the subgraph of  $G$  induced by the vertices of  $U$ . For any  $k \in \mathbb{N}$ , we write  $[k]$  as shorthand for  $\{1, \dots, k\}$ , and  $V^{(k)}$  for the set of all subsets of  $V$  of size exactly  $k$ , while  $V^k$  denotes the set of  $k$ -tuples of elements of  $V$ . A *permutation* on  $[k]$  is a bijection  $[k] \rightarrow [k]$ .

Two graphs  $G$  and  $H$  are *isomorphic*, written  $G \cong H$ , if there exists a bijection  $\theta : V(G) \rightarrow V(H)$  so that, for all  $u, v \in V(G)$ , we have  $\theta(u)\theta(v) \in E(H)$  if and only if  $uv \in E(G)$ ;  $\theta$  is said to be an *isomorphism* from  $G$  to  $H$ . An *automorphism* on  $G$  is an isomorphism from  $G$  to itself. We write  $\text{aut}(G)$  for the number of automorphisms of  $G$ . A *subgraph* of  $G$  is a graph of  $G$  obtained by possibly deleting vertices and/or edges from  $G$ ; a *spanning* subgraph is obtained by deleting edges only.  $H$  is said to be a subgraph of  $G$  if  $H$  is isomorphic to a subgraph of  $G$ .  $H$  is a *proper subgraph* of  $G$  if  $H$  is a subgraph of  $G$  but is not isomorphic to  $G$ .

If  $G$  is coloured by some colouring  $\omega : V \rightarrow [k]$ , we say that a subset  $U \subset V$  is *colourful* (under  $\omega$ ) if, for every  $i \in [k]$ , there exists exactly one vertex  $u \in U$  such that  $\omega(u) = i$ ; note that can only be achieved if  $|U| = k$ .

We will be considering labelled graphs, where a labelled graph is a pair

$(H, \pi)$  such that  $H$  is a graph and  $\pi : [|V(H)|] \rightarrow V(H)$  is a bijection. Given a graph  $G = (V, E)$  and a  $k$ -tuple of vertices  $(v_1, \dots, v_k)$ ,  $G[v_1, \dots, v_k]$  denotes the labelled graph  $(H, \pi)$  where  $H = G[\{v_1, \dots, v_k\}]$  and  $\pi(i) = v_i$  for each  $i \in [k]$ . If  $\mathcal{H}$  is a set of labelled graphs, we set  $\mathcal{H}^H = \{(H', \pi') \in \mathcal{H} : H' \cong H\}$ .

Given two graphs  $G$  and  $H$ , a *strong embedding* of  $H$  in  $G$  is an injective mapping  $\theta : V(H) \rightarrow V(G)$  such that, for any  $u, v \in V(H)$ ,  $\theta(u)\theta(v) \in E(G)$  if and only if  $uv \in E(H)$ . We denote by  $\#\text{StrEmb}(H, G)$  the number of strong embeddings of  $H$  in  $G$ . If  $\mathcal{H}$  is a class of labelled graphs, we set

$$\begin{aligned} \#\text{StrEmb}(\mathcal{H}, G) = \\ |\{\theta : [k] \rightarrow V(G) \quad : \quad \theta \text{ is injective and } \exists (H, \pi) \in \mathcal{H} \text{ such that} \\ \theta(i)\theta(j) \in E(G) \iff \pi(i)\pi(j) \in E(H)\}|. \end{aligned}$$

If  $G$  is also equipped with a  $k$ -colouring  $\omega$ , where  $|V(H)| = k$ , we write  $\#\text{ColStrEmb}(H, G, \omega)$  for the number of strong embeddings of  $H$  in  $G$  such that the image of  $V(H)$  is colourful under  $\omega$ . Similarly, we set

$$\begin{aligned} \#\text{ColStrEmb}(\mathcal{H}, G, \omega) = \\ |\{\theta : [k] \rightarrow V(G) \quad : \quad \theta \text{ is injective, } \exists (H, \pi) \in \mathcal{H} \text{ such that} \\ \theta(i)\theta(j) \in E(G) \iff \pi(i)\pi(j) \in E(H), \\ \text{and } \theta([k]) \text{ is colourful under } \omega\}|. \end{aligned}$$

We can alternatively consider unlabelled embeddings of  $H$  in  $G$ . In this context we write  $\#\text{SubInd}(H, G)$  for the number of subsets  $U \subseteq V(G)^{(|H|)}$  such that  $G[U] \cong H$ . Note that  $\#\text{SubInd}(H, G) = \#\text{StrEmb}(H, G) / \text{aut}(H)$ . If  $\mathcal{H}$  is a class of labelled graphs, we set

$$\#\text{SubInd}(\mathcal{H}, G) = |\{U \subseteq V(G) : \exists (H, \pi) \in \mathcal{H} \text{ such that } G[U] \cong H\}|.$$

Once again, we can also consider the case in which  $G$  is equipped with a  $k$ -colouring  $\omega$ . In this case  $\#\text{ColSubInd}(H, G, \omega)$  is the number of colourful subsets  $U$  such that  $G[U] \cong H$ , and

$$\begin{aligned} \#\text{ColSubInd}(\mathcal{H}, G, \omega) = |\{U \subseteq V(G) \quad : \quad \exists (H, \pi) \in \mathcal{H} \text{ such that } G[U] \cong H, \\ \text{and } U \text{ is colourful under } \omega\}|. \end{aligned}$$

Finally, we write  $\#\text{Clique}_k(G)$  as shorthand for  $\#\text{SubInd}(K_k, G)$ , where  $K_k$  denotes a clique on  $k$  vertices.

## 1.2 Parameterised Counting Complexity

In this section, we introduce key notions from parameterised counting complexity, which we will use in the rest of the paper.

As when considering the complexity of parameterised decision problems, an algorithm to solve a parameterised counting problem is considered to be efficient if, on input of size  $n$  with parameter  $k$ , its running time is bounded by  $f(k)n^{O(1)}$ , where  $f$  is any computable function of  $k$ ; problems (whether decision or counting problems) admitting such a *fpt-algorithm* belong to the class FPT and are said to be *fixed parameter tractable*.

To understand the intractability of parameterised counting problems, Flum and Grohe [10] introduce two kinds of reductions between such problems.

**Definition.** Let  $(\Pi, \kappa)$  and  $(\Pi', \kappa')$  be parameterised counting problems.

1. An *fpt parsimonious reduction* from  $(\Pi, \kappa)$  to  $(\Pi', \kappa')$  is an algorithm that computes, for every instance  $I$  of  $\Pi$ , an instance  $I'$  of  $\Pi'$  in time  $f(k) \cdot |I|^c$  such that  $\kappa'(I') \leq g(\kappa(I))$  and

$$\Pi(I) = \Pi'(I')$$

(for computable functions  $f, g : \mathbb{N} \rightarrow \mathbb{N}$  and a constant  $c \in \mathbb{N}$ ). In this case we write  $(\Pi, \kappa) \leq_{\text{pars}}^{\text{fpt}} (\Pi', \kappa')$ .

2. An *fpt Turing reduction* from  $(\Pi, \kappa)$  to  $(\Pi', \kappa')$  is an algorithm  $A$  with an oracle to  $\Pi'$  such that

- (a)  $A$  computes  $\Pi$ ,
- (b)  $A$  is an *fpt-algorithm* with respect to  $\kappa$ , and
- (c) there is a computable function  $g : \mathbb{N} \rightarrow \mathbb{N}$  such that for all oracle queries “ $\Pi'(y) = ?$ ” posed by  $A$  on input  $x$  we have  $\kappa'(y) \leq g(\kappa(x))$ .

In this case we write  $(\Pi, \kappa) \leq_{\text{T}}^{\text{fpt}} (\Pi', \kappa')$ .

Using these notions, Flum and Grohe introduce a hierarchy of parameterised counting complexity classes,  $\#W[t]$ , for  $t \geq 1$ ; this is the analogue of the W-hierarchy for parameterised decision problems. In order to define this hierarchy, we need some more notions related to satisfiability problems.

The definition of levels of the hierarchy uses the following problem.

$p\text{-}\#\text{WD}_\psi$

*Input:* A structure  $\mathcal{A}$  and  $k \in \mathbb{N}$ .

*Parameter:*  $k$ .

*Question:* How many relations  $S \subseteq A^s$  of cardinality  $|S| = k$  are such that  $\mathcal{A} \models \psi(S)$  (where  $A$  is the universe of  $\mathcal{A}$ )?

If  $\Psi$  is a class of first-order formulas, then  $p\text{-}\#\text{WD-}\Psi$  is the class of all problems  $p\text{-}\#\text{WD}_\psi$  where  $\psi \in \Psi$ . The classes of first-order formulas  $\Sigma_t$  and  $\Pi_t$ , for  $t \geq 0$ , are defined inductively. Both  $\Sigma_0$  and  $\Pi_0$  denote the class of quantifier-free formulas, while, for  $t \geq 1$ ,  $\Sigma_t$  is the class of formulas

$$\exists x_1 \dots \exists x_i \psi,$$

where  $\psi \in \Pi_{t-1}$ , and  $\Pi_t$  is the class of formulas

$$\forall x_1 \dots \forall x_i \psi,$$

where  $\psi \in \Sigma_{t-1}$ . We are now ready to define the classes  $\#W[t]$ , for  $t \geq 1$ .

**Definition** ([10, 11]). *For  $t \geq 1$ ,  $\#W[t]$  is the class of all parameterised counting problems that are fpt parsimonious reducible to  $p\text{-}\#\text{WD-}\Pi_t$ .*

Just as it is considered to be very unlikely that  $W[1] = \text{FPT}$ , it is very unlikely that there exists an algorithm running in time  $f(k)n^{O(1)}$  for any problem that is hard for the class  $\#W[1]$ ; hardness of a problem can be shown using either of the forms of reductions defined above. We have already mentioned the  $\#W[1]$ -complete problem  $p\text{-}\#\text{CLIQUE}$  (shown to be hard in [10]), which is defined formally as follows.

$p\text{-}\#\text{CLIQUE}$

*Input:* A graph  $G = (V, E)$ , and  $k \in \mathbb{N}$ .

*Parameter:*  $k$ .

*Question:* How many  $k$ -vertex cliques are there in  $G$ ?

When considering approximation algorithms for parameterised counting problems, an “efficient” approximation scheme is an FPTRAS, as introduced by Arvind and Raman [3]; this is the analogue of a FPRAS (fully polynomial randomised approximation scheme) in the parameterised setting.

**Definition.** An *FPTRAS* for a parameterised counting problem  $\Pi$  with parameter  $k$  is a randomised approximation scheme that takes an instance  $I$  of  $\Pi$  (with  $|I| = n$ ), and real numbers  $\epsilon > 0$  and  $0 < \delta < 1$ , and in time  $f(k) \cdot g(n, 1/\epsilon, \log(1/\delta))$  (where  $f$  is any function, and  $g$  is a polynomial in  $n$ ,  $1/\epsilon$  and  $\log(1/\delta)$ ) outputs a rational number  $z$  such that

$$\mathbb{P}[(1 - \epsilon)\Pi(I) \leq z \leq (1 + \epsilon)\Pi(I)] \geq 1 - \delta.$$

Later in the paper, we will also consider the parameterised complexity of decision problems. This will involve the parameterised complexity class  $W[1]$ , the first level of the  $W$ -hierarchy (originally introduced by Downey and Fellows [8]), which is the decision equivalent of the  $\#W$ -hierarchy. The levels of the  $W$ -hierarchy can be defined in a similar way to those of the  $\#W$ -hierarchy, using the following problem.

**WD $_{\psi}$**

*Input:* A structure  $\mathcal{A}$  and  $k \in \mathbb{N}$ .

*Parameter:*  $k$ .

*Question:* Is there  $S \subseteq A^s$  of cardinality  $|S| = k$  such that  $\mathcal{A} \models \psi(S)$  (where  $A$  is the universe of  $\mathcal{A}$ )?

The classes  $W[t]$ , for  $t \geq 1$ , are then defined as follows.

**Definition** ([11]). For  $t \geq 1$ ,  $W[t]$  is the class of all parameterised problems that are *fpt-reducible* to  $WD\text{-}\Pi_t$ .

For the formal definition of fpt-reductions and other notions from the theory of parameterised complexity, we refer the reader to [11]. When proving  $W[1]$ -hardness of various problems later in the paper, we will use reductions from the problem  $p\text{-CLIQUE}$ , shown to be  $W[1]$ -complete in [8], and formally defined as follows:

**$p\text{-CLIQUE}$**

*Input:* A graph  $G = (V, E)$ , and  $k \in \mathbb{N}$ .

*Parameter:*  $k$ .

*Question:* Does  $G$  contain a  $k$ -vertex clique?



## 2 The model

The classes of counting problems we consider fall within the scope of the general model introduced in [13]; this model describes parameterised counting problems in which the goal is to count labelled subgraphs with particular properties. We repeat the definition here for completeness, as well as the extension to multicolour subgraph counting problems defined in [14], before extending it to the corresponding decision problems.

Let  $\Phi$  be a family  $(\phi_1, \phi_2, \dots)$  of functions  $\phi_k : \{0, 1\}^{\binom{k}{2}} \rightarrow \{0, 1\}$ , such that the function mapping  $k \mapsto \phi_k$  is computable. Let  $(i_1, \dots, i_{\binom{k}{2}})$  be a fixed ordering of all pairs in  $[k]^{\binom{k}{2}}$ . For any labelled graph  $(H, \pi)$  we define

$$\phi_k(H, \pi) = \phi_k(e_{i_1}^{(H, \pi)}, e_{i_2}^{(H, \pi)}, \dots, e_{i_{\binom{k}{2}}}^{(H, \pi)}),$$

where  $e_{\{j, l\}}^{(H, \pi)} \in \{0, 1\}$  and  $e_{\{j, l\}} = 1$  if and only if  $\pi(j)\pi(l) \in E(H)$ .

The general problem is then defined as follows.

*p*-#INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ )

*Input:* A graph  $G = (V, E)$  and an integer  $k$ .

*Parameter:*  $k$ .

*Question:* What is the cardinality of the set  $\{(v_1, \dots, v_k) \in V^k : \phi_k(G[v_1, \dots, v_k]) = 1\}$ ?

It was argued in [13] that this problem lies in #W[1]:

**Proposition 2.1.** *For any  $\Phi$ , the problem *p*-#INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) belongs to #W[1].*

Much of this paper will be concerned with the multicolour version of the general problem, introduced in [14] (and also known to belong to #W[1]).

*p*-#MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ )

*Input:* A graph  $G = (V, E)$ , an integer  $k$  and colouring  $f : V \rightarrow [k]$ .

*Parameter:*  $k$ .

*Question:* What is the cardinality of the set  $\{(v_1, \dots, v_k) \in V^k : \phi_k(G[v_1, \dots, v_k]) = 1 \text{ and } \{f(v_1), \dots, f(v_k)\} = [k]\}$ ?

In Section 4 we will also consider the decision versions of both the multicolour and uncoloured variants of the problem, which are defined as follows.

*p*-INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ )

*Input:* A graph  $G = (V, E)$ , and an integer  $k$ .

*Parameter:*  $k$ .

*Question:* Is there any tuple  $(v_1, \dots, v_k) \in V^k$  such that  $\phi_k(G[v_1, \dots, v_k]) = 1$ ?

*p*-MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ )

*Input:* A graph  $G = (V, E)$ , an integer  $k$  and colouring  $f : V \rightarrow [k]$ .

*Parameter:*  $k$ .

*Question:* Is there any tuple  $(v_1, \dots, v_k) \in V^k$  such that  $\phi_k(G[v_1, \dots, v_k]) = 1$  and  $\{f(v_1), \dots, f(v_k)\} = [k]$ ?

It is straightforward to verify that both these problems belong to the parameterised complexity class  $W[1]$ .

In a number of our results below, we will be concerned with the case in which  $\Phi$  has certain special features; we now define two such features of interest. First of all, we say that  $\Phi$  is monotone if and only if, for each  $k \in \mathbb{N}$ , whenever  $x = (x_1, \dots, x_{\binom{k}{2}}), y = (y_1, \dots, y_{\binom{k}{2}}) \in \{0, 1\}^m$  satisfy  $x_i \leq y_i$  for all  $1 \leq i \leq \binom{k}{2}$ , then

$$\phi(x) = 1 \implies \phi(y) = 1.$$

Any monotone property  $\Phi$  is uniquely defined by the *minimal labelled graphs satisfying  $\phi_k$* , for each  $k$ : a labelled graph  $(H, \pi)$  is a minimal labelled graph satisfying  $\phi_k$  if  $\phi_k(H, \pi) = 1$  but, for every proper spanning subgraph  $H'$  of  $H$  (that is, a graph obtained from  $H$  by deleting at least one edge but no vertices),  $\phi_k(H', \pi) = 0$ . We say that  $H$  is a *minimal unlabelled graph satisfying  $\phi_k$*  if there exists a labelling  $\pi$  such that  $\phi_k(H, \pi) = 1$ , but for every proper spanning subgraph  $H'$  of  $H$  and *every* labelling  $\pi'$ , we have  $\phi_k(H', \pi') = 0$ .

We say that a monotone property  $\Phi$  is *nice* if, for each  $k \in \mathbb{N}$  and unlabelled subgraph  $H$  on  $k$  vertices,  $H$  is a minimal unlabelled subgraph satisfying  $\phi_k$  if and only if there exists a labelling  $\pi$  such that  $(H, \pi)$  is a minimal labelled subgraph satisfying  $\phi_k$ . Observe that all of the specific problems previously studied in the literature (including *p*-#CLIQUE, *p*-#PATH,

$p$ -#CYCLE,  $p$ -#MATCHING,  $p$ -#CONNECTED INDUCED SUBGRAPH) can be expressed as instances of  $p$ -#INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) where  $\Phi$  is a nice property.

## 2.1 The existing complexity landscape

In this section we review some important facts about the complexities of problems described by the various models introduced above.

### Relative complexities of problem variants

Earlier in this section, we introduced both the uncoloured and multicolour versions of our general subgraph counting problem, as well as the corresponding pair of decision problems. If we consider separately the complexities of exact and approximate counting, the different versions of the model give rise to six distinct complexity questions for any given property  $\Phi$ : for both the uncoloured and multicolour versions, we can ask the following three questions.

1. Is the decision version in FPT?
2. Is there an FPTRAS for the counting version?
3. Is the exact counting problem in FPT?

The answers to some of these questions have implications for some of the others. First of all, it is clear that if the exact counting problem is in FPT for some property  $\Phi$ , then there is trivially an FPTRAS for the problem and also the corresponding decision problem must be in FPT.

To consider the relationship between decision and approximate counting, we need a further definition. The class *randomised FPT* is the class of parameterised decision problems for which there is a randomised algorithm with the following properties:

1. the algorithm runs in time bounded by  $f(k) \cdot n^{O(1)}$  on input of size  $n$  with parameter  $k$ , where  $f$  is some computable function,
2. if the correct answer is “NO”, the algorithm always outputs “NO”.
3. if the correct answer is “YES”, the algorithm outputs “YES” with probability strictly greater than  $\frac{1}{2}$ .

In the following proposition, we show that the existence of an FPTRAS for a counting problem in our model implies that the corresponding decision problem belongs to randomised FPT.

**Proposition 2.2.** *Let  $\Phi$  be a family  $(\phi_1, \phi_2, \dots)$  of functions  $\phi_k : \{0, 1\}^{\binom{k}{2}} \rightarrow \{0, 1\}$ , such that the function mapping  $k \mapsto \phi_k$  is computable. Then, if there is a FPTRAS for  $p$ -#INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ), it follows that  $p$ -INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) is in randomised FPT. Similarly, if there is a FPTRAS for  $p$ -#MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ), it follows that  $p$ -MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) is in randomised FPT.*

*Proof.* Suppose that there exists an FPTRAS for  $p$ -#INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ). Running this randomised approximation algorithm with  $\epsilon = 1/2$  and  $\delta = \frac{1}{2n+1}$  immediately gives a randomised fixed parameter tractable decision algorithm  $A$  with two-sided error probability strictly smaller than  $\frac{1}{2n}$ : if the answer to the decision problem is “NO” then the approximation algorithm must output “0” with probability at least  $\frac{1}{2n+1}$ , whereas if the answer to the decision problem is “YES” then the approximation algorithm must output a value of at least  $1/2$  with probability at least  $\frac{1}{2n+1}$ . It remains to show that this two-sided error can be reduced to a one-sided error, without increasing the probability of error too much.

We do this using standard techniques, similar to those employed to show that certain problems (so-called “self-reducible” problems) in the class BPP are in fact contained in RP; the idea is that before outputting the answer “YES” we will search for a witness to the fact that the input is indeed a yes-instance. We begin by running our randomised decision algorithm  $A$ ; if the output of  $A$  is “NO” then we return “NO”. If, on the other hand, the output of  $A$  is “YES” we begin our search for a witness.

Suppose our input graph is  $G$ , and that the parameter is  $k$ . The first step in the search for a witness is to delete an arbitrary vertex  $v$  from our input graph  $G$ , and running  $A$  on the resulting graph. If the output of  $A$  on this new graph is “YES”, we repeat the deletion process (but always ensuring that we do not delete a vertex we have previously marked as “necessary”); if the output is “NO” then we mark the vertex  $v$  as “necessary”, delete a vertex that is not yet marked as necessary, and continue. This process terminates when either:

1.  $k + 1$  vertices have been marked as necessary, in which case we return “NO”, or

2. there are exactly  $k$  vertices remaining, in which case we check (in time depending only on  $k$ ) whether  $\phi_k$  is true on any tuple formed of the remaining vertices; if so we return “YES”, and otherwise we return “NO”.

It is clear that this witness search process is fixed parameter tractable, and moreover that it will only return “YES” if we do indeed have a yes-instance. To bound the probability of returning “NO” if we actually have a yes-instance, observe that we call the algorithm  $A$  at most  $n$  times (where  $n$  is the number of vertices in  $G$ ), so by the union bound the probability that the algorithm will return the incorrect answer on at least one of these calls is at most  $\frac{n}{2n+1} < \frac{1}{2}$ ; if  $A$  returns the correct answer on all of the calls then we are sure to return “YES” on any yes-instance. Thus the procedure described does indeed give a randomised fpt-algorithm for  $p$ -INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ).

The corresponding result for the multicolour versions of the problems follows by exactly the same argument.  $\square$

Subject to the widely held assumption that  $W[1] \not\subseteq$  randomised FPT, this gives the following immediate corollary.

**Corollary 2.3.** *Assume that  $W[1] \not\subseteq$  randomised FPT, and let  $\Phi$  be a family  $(\phi_1, \phi_2, \dots)$  of functions  $\phi_k : \{0, 1\}^{\binom{k}{2}} \rightarrow \{0, 1\}$ , such that the function mapping  $k \mapsto \phi_k$  is computable. Then, if  $p$ -INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) is  $W[1]$ -complete under fpt many-one reductions, there is no FPTRAS for  $p$ -#INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ). Similarly, if  $p$ -MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) is  $W[1]$ -complete, there is no FPTRAS for  $p$ -#MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ).*

The relationships between the complexities of multicolour and uncoloured versions of the problems are somewhat more complicated. For the case of exact counting, the following result was proved in [13].

**Lemma 2.4.** *For any family  $\Phi$ , we have  $p$ -#MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ )  $\leq_T^{fpt}$   $p$ -#INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ).*

However, this relationship is reversed for both decision and approximate counting: the use of colour-coding techniques (as described in [2] and [1]) gives the following two results.

**Proposition 2.5.** *For any family  $\Phi$ , if  $p$ -MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) is in FPT then so is  $p$ -INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ).*

**Proposition 2.6.** *For any family  $\Phi$ , if there is an FPTRAS for  $p$ -#MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) then there is an FPTRAS for  $p$ -#INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ).*

The complexity implications from the preceding results are summarised in Figure 1; an arrow from problem A to problem B implies that (subject to standard assumptions in parameterised complexity theory), if problem A is efficiently solvable (either it is in FPT, or it admits an FPTRAS, as appropriate), then there must also be an efficient algorithm for problem B.

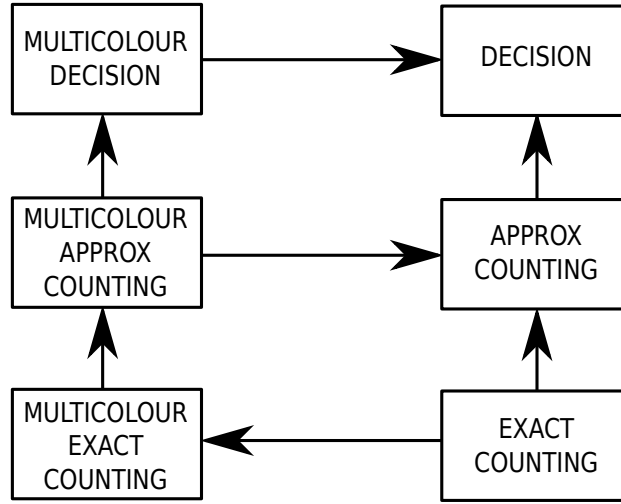


Figure 1: Summary of the complexity implications between the problems considered

### A useful example

In Figure 1 above, it is natural to wonder whether some of the problems variants for any given  $\Phi$  might in fact be equivalent under fpt-Turing reductions. Here we give an example which demonstrates that, for certain pairs of problem variants, this is not the case: in particular, the converses of Propositions 2.5 and 2.6 do not hold. For our example,  $\Phi = (\phi_1, \phi_2, \dots)$  is defined so that

$\phi_k(G[v_1, \dots, v_k])$  is true if and only if the vertices  $v_1, \dots, v_k$  induce either a clique or independent set in  $G$ . We call the decision problem in this case  $p$ -CLIQUE OR INDEPENDENT SET, and the counting version  $p$ -#CLIQUE OR INDEPENDENT SET.

The following result was proved in [3], using the fact from Ramsey Theory that any graph on  $n$  vertices is guaranteed to contain either a clique or independent set on  $k$  vertices, provided that  $n$  is sufficiently large compared with  $k$ .

**Lemma 2.7** ([3]).  *$p$ -CLIQUE OR INDEPENDENT SET is in FPT.*

Using a similar construction to that exploited in [3] to demonstrate the #W[1]-hardness of  $p$ -#CLIQUE OR INDEPENDENT SET, it is straightforward to prove that the multicolour decision problem is W[1]-complete.

**Lemma 2.8.**  *$p$ -MULTICOLOUR CLIQUE OR INDEPENDENT SET is W[1]-complete.*

*Proof.* We proceed by means of a reduction from  $p$ -MULTICOLOUR CLIQUE, shown to be W[1]-complete in [9]. Let  $G$  with colouring  $f$  be the input to an instance of  $p$ -MULTICOLOUR CLIQUE; without loss of generality we may assume that  $f$  colours the vertices of  $G$  with colours  $\{1, \dots, k\}$ . We now construct  $G'$  from  $G$  by adding a new vertex  $v$  adjacent to every vertex of  $G$ ; we extend the colouring  $f$ , to give a colouring  $f'$  of the vertices of  $G'$ , by setting  $f(v) = k + 1$ . Any colourful subset of vertices in  $G'$  with colouring  $f'$  must therefore contain  $v$ , but as  $v$  is adjacent to all other vertices it cannot be part of any  $k + 1$ -independent set. Thus, there is a colourful clique or independent set in  $G'$  (with colouring  $f'$ ) if and only if there is a colourful clique in  $G$  (with colouring  $f$ ).  $\square$

As a consequence, it is clear that there is unlikely to be an FPTRAS for  $p$ -#MULTICOLOUR CLIQUE OR INDEPENDENT SET. However, we now demonstrate that the uncoloured counting problem is efficiently approximable, answering an open question from [3]. We begin with an auxiliary result, proved in [14].

**Proposition 2.9** ([14]). *Let  $G = (V, E)$  be an  $n$ -vertex graph, where  $n \geq 2^k$ . Then the number of  $k$ -vertex subsets  $U \subset V$  such that  $U$  induces either a clique or independent set in  $G$  is at least*

$$\frac{(2^k - k)!}{(2^k)!} \frac{n!}{(n - k)!}.$$

Using this fact, we can now demonstrate the existence of an FPTRAS for  $p$ -#CLIQUE OR INDEPENDENT SET.

**Lemma 2.10.** *There exists an FPTRAS for  $p$ -#CLIQUE OR INDEPENDENT SET*

*Proof.* We obtain an approximation to the total number  $N$  of  $k$ -vertex cliques and independent sets in our graph using a simple random sampling algorithm. At each step, a set of  $k$  vertices is picked uniformly at random from all such sets, and then we check whether this set induces a clique or independent set; the sampling and checking can clearly be performed in time  $f(k) \cdot n^{O(1)}$  for a computable function  $f$ . To obtain a good estimate of the total number of cliques and independent sets in the entire graph, we repeat this sampling process  $t$  times (for some value of  $t$  to be determined), and compute the proportion  $p$  of our sampled sets which did induce either a clique or independent set. We then output as our approximation  $p \binom{n}{k}$ .

The value of  $t$  must be chosen to be large enough that the probability that  $|p \binom{n}{k} - N| \leq \epsilon N$  is at least  $1 - \delta$ . However, we know that the probability of any randomly chosen subset of  $k$  vertices inducing either a clique or an independent set is at least

$$\frac{(2^k - k)!}{(2^k)!} \frac{n!}{(n - k)!} \cdot \frac{1}{\binom{n}{k}} = \frac{(2^k - k)!k!}{(2^k)!}.$$

Thus it is straightforward to verify (using, for example, a Chernoff bound) that the number of trials required is bounded by a  $g(k) \cdot q(\epsilon^{-1}, \log(\delta^{-1}))$ , where  $g$  is a computable function and  $q$  is a polynomial in  $\epsilon^{-1}$  and  $\log(\delta^{-1})$ , as required to satisfy the definition of an FPTRAS.  $\square$

## 2.2 Graph minors and Treewidth

The graph parameter treewidth will play a crucial role in the remainder of the paper. We say that  $(T, \mathcal{D})$  is a *tree decomposition* of  $G$  if  $T$  is a tree and  $\mathcal{D} = \{\mathcal{D}(t) : t \in V(T)\}$  is a collection of non-empty subsets of  $V(G)$  (or *bags*), indexed by the nodes of  $T$ , satisfying:

1.  $V(G) = \bigcup_{t \in V(T)} \mathcal{D}(t)$ ,



2. for every  $e = uv \in E(G)$ , there exists  $t \in V(T)$  such that  $u, v \in \mathcal{D}(t)$ ,
3. for every  $v \in V(G)$ , if  $T(v)$  is defined to be the subgraph of  $T$  induced by nodes  $t$  with  $v \in \mathcal{D}(t)$ , then  $T(v)$  is connected.

The *width* of the tree decomposition  $(T, \mathcal{D})$  is defined to be  $\max_{t \in V(T)} |\mathcal{D}(t)| - 1$ , and the *treewidth* of  $G$ , written  $\text{tw}(G)$ , is the minimum width over all tree decompositions of  $G$ .

Another key concept in the following sections is that of *graph minors*; this extends the idea of subgraphs. The graph  $H$  is said to be a *minor* of  $G$  if there is a mapping  $m : V(H) \rightarrow \mathcal{P}(V(G))$ , mapping vertices of  $H$  to sets of vertices from  $G$ , satisfying:

1. for all  $u \in V(H)$ ,  $G[m(u)]$  is connected,
2. for  $u, v \in V(H)$  with  $u \neq v$ ,  $m(u) \cap m(v) = \emptyset$ , and
3. for all  $uv \in E(H)$ , there exist  $u' \in m(u)$  and  $v' \in m(v)$  such that  $u'v' \in E(G)$ .

The concepts of treewidth and graph minors are combined in the celebrated Excluded Grid Theorem, proved by Robertson and Seymour.

**Theorem 2.11** (Excluded Grid Theorem [16]). *There is a computable function  $w : \mathbb{N} \rightarrow \mathbb{N}$  such that the  $(k \times k)$  grid is a minor of every graph of treewidth at least  $w(k)$ .*

## Complexity results involving treewidth

A number of specific results concerning the complexity of problems described by our model have already been mentioned. Here we state more known results for such problems, all of which involve the concept of treewidth.

The first, rewritten in the language of our model, states that problem of deciding whether a graph  $G$  contains a copy (not necessarily induced) of a given  $k$ -vertex graph  $H$  of bounded treewidth is in FPT.

**Theorem 2.12** ([15]). *Let  $\Phi$  be a family  $(\phi_1, \phi_2, \dots)$  of functions  $\phi_k : \{0, 1\}^{\binom{k}{2}} \rightarrow \{0, 1\}$ , such that the function mapping  $k \mapsto \phi_k$  is computable. Suppose that  $\Phi$  is monotone, and that for each  $k \in \mathbb{N}$  there is a unique minimal labelled graph  $(H_k, \pi_k)$  satisfying  $\phi_k$ . If there exists some constant  $t \in \mathbb{N}$  such that, for each  $k$ ,  $\text{tw}(H_k) \leq t$ , then  $p$ -INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) and  $p$ -MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) are in FPT.*

The following more general result follows as an easy corollary.

**Corollary 2.13.** *Let  $\Phi = (\phi_1, \phi_2, \dots)$  be a monotone property, and suppose there exists a positive integer  $t$  such that, for each  $\phi_k$ , all minimal labelled  $k$ -vertex graphs  $(H, \pi)$  such that  $\phi_k(H) = 1$  satisfy  $\text{tw}(H) \leq t$ . Then  $p$ -INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) and  $p$ -MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) are in FPT.*

It is not hard to show that the same is true provided that all minimal *unlabelled* subgraphs with the property have bounded treewidth.

**Corollary 2.14.** *Let  $\Phi = (\phi_1, \phi_2, \dots)$  be a monotone property, and suppose there exists a positive integer  $t$  such that, for each  $\phi_k$ , all minimal unlabelled  $k$ -vertex graphs  $(H, \pi)$  such that  $\phi_k(H) = 1$  satisfy  $\text{tw}(H) \leq t$ . Then  $p$ -INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) and  $p$ -MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) are in FPT.*

*Proof.* We will define a family  $\Phi' = (\phi'_1, \phi'_2, \dots)$  of monotone functions  $\phi'_k : \{0, 1\}^{\binom{k}{2}} \rightarrow \{0, 1\}$  (which may be constructed from  $\Phi$  in time  $f(k)n^{O(1)}$  for some computable function  $f$ ) such that for every  $k$  all minimal *labelled* subgraphs satisfying  $\phi'_k$  have treewidth at most  $t$ , and argue that a graph  $G$  contains a  $k$ -vertex subgraph satisfying  $\phi'_k$  if and only if  $G$  contains a  $k$ -vertex subgraph satisfying  $\phi_k$ . The result will then follow from Corollary 2.13.

Suppose that the minimal labelled subgraphs satisfying  $\phi_k$  are precisely  $(H_1^k, \pi_1^k), \dots, (H_{r_k}^k, \pi_{r_k}^k)$ . If each  $H_i^k$  is also a minimal unlabelled subgraph satisfying  $\phi_k$ , we are done with  $\phi'_k = \phi_k$ , so suppose without loss of generality that there exists  $1 \leq s_k \leq r_k$  such that the minimal unlabelled subgraphs satisfying  $\phi_k$  are precisely  $H_1^k, \dots, H_{s_k}^k$ . Now set the minimal labelled subgraphs satisfying  $\phi'_k$  to be  $(H_1^k, \pi_1^k), \dots, (H_{s_k}^k, \pi_{s_k}^k)$ ; as  $\Phi'$  is monotone, this uniquely specifies  $\phi'_k$ . It is clear from the definition that if  $\phi'_k(G[v_1, \dots, v_k]) = 1$  then so also  $\phi_k(G[v_1, \dots, v_k]) = 1$ . Conversely, suppose that  $\phi_k(G[v_1, \dots, v_k]) = 1$ . If  $G[v_1, \dots, v_k] \in \{(H_1^k, \pi_1^k), \dots, (H_{s_k}^k, \pi_{s_k}^k)\}$  then we are done, so it must be that, for some  $i \in \{s_k + 1, \dots, r_k\}$ ,  $G[v_1, \dots, v_k] = (H_i^k, \pi_i^k)$ . Since  $H_i^k$  is not a minimal unlabelled subgraph satisfying  $\phi_k$ , there must be some  $1 \leq j \leq s_k$  such that  $H_j^k$  is a proper spanning subgraph of  $H_i^k$ ; in other words there exists a permutation  $\sigma : [k] \rightarrow [k]$  such that, for all  $a, b \in [k]$ , we have  $\pi_i^k(a)\pi_i^k(b) \in E(H_i^k)$  whenever  $\sigma(\pi_j^k(a))\sigma(\pi_j^k(b)) \in E(H_j^k)$ . It follows immediately that  $\phi'_k(G[v_{\sigma(1)}, \dots, v_{\sigma(k)}]) = 1$ , as required.  $\square$

The counting version of the problem for any  $\Phi$  for which the minimal labelled satisfying subgraphs have bounded treewidth was shown to admit an FPTRAS in [13]; it follows immediately from the proof given in [13] that the same is true for the multicolour version in this situation.

**Theorem 2.15** ([13]). *Let  $\Phi = (\phi_1, \phi_2, \dots)$  be a monotone property, and suppose there exists a positive integer  $t$  such that, for each  $\phi_k$ , we have  $\text{tw}(H) \leq t$  whenever  $(H, \pi)$  is a minimal labelled subgraph satisfying  $\phi_k$ . Then there is an FPTRAS for  $p$ -#INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ), and an FPTRAS for  $p$ -#MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ).*

## 2.3 Examples of problems covered by our results

Some obvious examples of problems, from the model introduced above, to which our intractability results apply come from properties that in some way depend on the treewidth of the graph. For example, if we set  $\phi_k(H, \pi) = 1$  if and only if  $H$  has treewidth at least  $\theta(k)$ , where  $\theta$  is any function such that  $\theta(k) \rightarrow \infty$  as  $k \rightarrow \infty$ , then it is clear that there cannot be any constant  $t$  such that, for all  $k$ , all minimal (unlabelled) subgraphs satisfying  $\phi_k$  have treewidth at most  $t$  (indeed, for any constant  $t$ , there will exist  $k \in \mathbb{N}$  such that *all* minimal subgraphs satisfying  $\phi_k$  have treewidth greater than  $t$ ). The same is true if, for example, we set  $\phi_k(H, \pi) = 1$  if and only if the chromatic number of  $H$  is at least  $\theta(k)$  (where again  $\theta(k) \rightarrow \infty$  as  $k \rightarrow \infty$ ), since any graph with treewidth at most  $t$  has chromatic number at most  $t + 1$ .

It is also clear that our results apply when  $\phi_k$  is true only if  $H$  contains a complete bipartite spanning subgraph. This category of problems includes the case in which  $\phi_k(H) = 1$  if and only if  $k$  is even and  $H \cong K_{\frac{k}{2}, \frac{k}{2}}$ , a problem whose parameterised complexity from the point of view of decision, in the uncoloured case, is a long-standing open problem.

Slightly less obvious examples of problems for which there is no  $t \in \mathbb{N}$  that bounds the treewidth of all minimal (unlabelled) elements satisfying  $\phi_k$ , for all  $k$ , arise when  $\phi_k$  is true on a  $k$ -vertex subgraph if and only if certain connectivity conditions are satisfied: while  $p$ -#CONNECTED INDUCED SUBGRAPH, studied in [13], is an example of a property for which all minimal elements do have bounded treewidth (since the minimal elements satisfying  $\phi_k$  are precisely all trees on  $k$  vertices), if we instead demand stronger connectivity conditions this is no longer the case. It is straightforward to verify

that the graph obtained by subdividing every edge of the  $k \times k$  grid exactly once (that is, replacing every edge of the grid with a new vertex adjacent to both endpoints of the original edge) has treewidth at least  $k$  and is also a minimally 2-connected graph, in the sense that it is 2-connected but deleting any edge will result in a graph that is no longer 2-connected. Similarly, the graph illustrated in Figure 2 has treewidth at least  $k$  (as it contains the  $k \times k$  grid as a minor) and is 3-connected, but deleting any edge will result in a graph that is no longer 3-connected (as such a subgraph will contain at least one vertex of degree at most 2, and so the graph cannot possibly be 3-connected). Both of these problems were already shown to be hard from the point of view of decision in the uncoloured case in [4], in which the authors proved that the problem of deciding whether a graph contains an  $r$ -connected induced subgraph on  $k$  vertices is  $W[1]$ -complete.

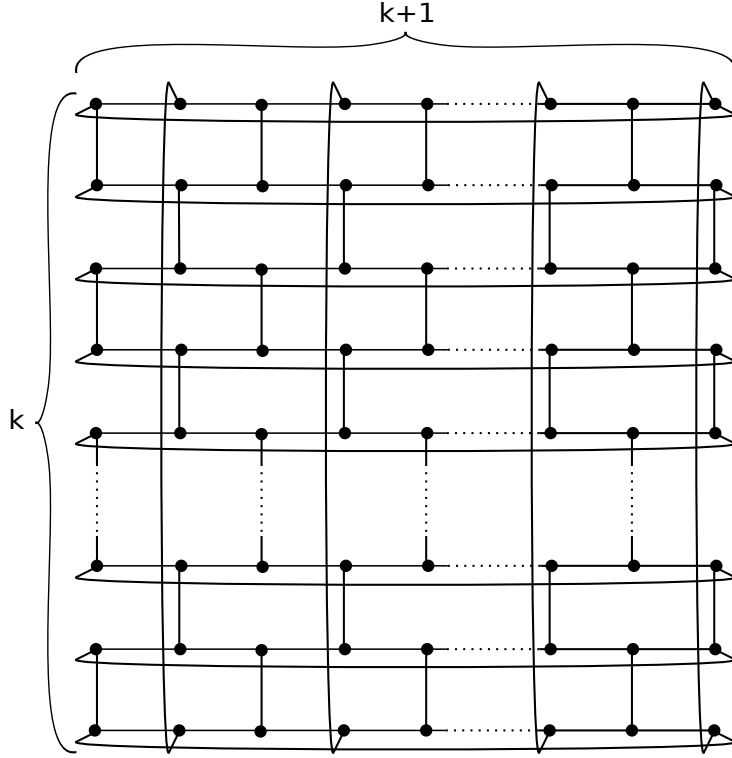


Figure 2: A minimally 3-connected graph with treewidth at least  $k$ .

### 3 A construction

In this section, we describe the construction that we will exploit to prove our complexity results in the following section. Let  $G$  be the graph in an instance of  $p$ -CLIQUE with parameter  $k$ , and let  $H$  be a graph which contains the  $k \times \binom{k}{2}$  grid as a minor. We then define a new coloured graph  $G_H$ , and a colouring  $f$  of its vertices with  $|V(H)|$  colours. In order to define our construction, it will be necessary to fix an arbitrary total order  $\prec$  on  $V(G)$ .

Let  $A$  denote the  $k \times \binom{k}{2}$  grid. We will label each of the  $k$  rows with a distinct element of the set  $\{1, \dots, k\}$ , and each of the  $\binom{k}{2}$  columns with a distinct unordered pair of elements from the same set. We then denote by  $a_{(i, \{j, l\})}$  the vertex in row  $i$  and column  $j$ .

Since the grid  $A$  is a minor of  $H$ , there exists a function  $m : V(A) \rightarrow \mathcal{P}(V(H))$  such that the image of each vertex in  $A$  induces a connected subgraph in  $H$ , distinct vertices of  $A$  map to disjoint subsets of  $V(H)$ , and for each edge  $aa' \in A$  there exist  $u \in m(a)$  and  $u' \in m(a')$  such that  $uu' \in E(H)$ . Note that the union of the image of  $m$  is not necessarily the whole of  $V(H)$ ; if not, then we denote by  $V'_H$  the vertices of  $V(H) \setminus \bigcup_{a \in V(A)} m(a)$ .

In our new graph  $G_H$ , we will have multiple copies of the subgraph  $H[m(a)]$  of  $H$  for each  $a \in V(A)$ , each one indexed by a pair consisting of a vertex and an edge from  $G$ . Specifically, for each  $a_{(i, \{j, l\})} \in V(A)$ , we will have a copy  $H_{(i, \{j, l\})}^{(v, e)}$  of  $H[m(a_{(i, \{j, l\})})]$  for each pair  $(v, e) \in V(G) \times E(G)$  that satisfies the condition

$$\text{if } i \in \{j, l\} \text{ then } v \text{ is incident with } e. \quad (1)$$

For  $u \in m(a_{(i, \{j, l\})})$ , we denote by  $u^{(v, e)}$  the corresponding vertex in  $H_{(i, \{j, l\})}^{(v, e)}$  (note that each  $u \in V(H)$  belongs to subgraphs  $H_{(i, \{j, l\})}$  for only a single value of  $(i, \{j, l\})$ ). For each  $(i, \{j, l\}) \in [k] \times [k]^{(2)}$ , we denote by  $V_{(i, \{j, l\})}$  the set of vertices

$$\begin{aligned} \{V(H_{(i, \{j, l\})}^{(v, e)}) : (v, e) \in V(G) \times E(G), \text{ and} \\ i \in \{j, l\} \implies v \text{ incident with } e\}. \end{aligned}$$

We now set

$$V(G_H) = \bigcup_{(i, \{j, l\}) \in [k] \times [k]^{(2)}} V_{(i, \{j, l\})} \cup V'_H.$$

That is, the vertex-set of  $G_H$  is made up of all the vertices in subgraphs  $H_{(i,\{j,l\})}^{(v,e)}$ , together with any vertices in  $V'_H$ .

We now define the colouring  $f$  of  $V(G_H)$ . First, fix a colouring  $\omega$  of  $V(H)$  which gives every vertex a different colour from the set  $\{1, \dots, |V(H)|\}$ . For any  $u \in V'_H$ , we now set  $f(u) = \omega(u)$ , while for every vertex  $u^{(v,e)} \in V(G_H)$  we set  $f(u^{(v,e)}) = \omega(u)$ .

To complete the definition of  $G_H$ , we now define its edge-set. Let  $u, w \in V(G_H)$ . Then  $uw \in E(G_H)$  if and only if the following conditions are satisfied:

1.  $\omega^{-1}f(u)\omega^{-1}f(w) \in E(H)$ ,
2. if  $u \in H_{(i,\{j,l\})}^{(v,e)}$  and  $w \in H_{(i',\{j',l'\})}^{(v',e')}$  then
  - (a) if  $i = i'$  then  $v = v'$ , and if  $i < i'$  then  $v \prec v'$ , and
  - (b) if  $\{j, l\} = \{j', l'\}$  then  $e = e'$ .

Note that the second condition implies that there are no edges between  $H_{(i,\{j,l\})}^{(v,e)}$  and  $H_{(i,\{j,l\})}^{(v',e')}$  for  $(v, e) \neq (v', e')$ .

Finally, observe that, given  $G$  and  $H$ , the graph  $G_H$  and its colouring  $f$  can clearly be constructed in time  $h(k)n^{O(1)}$ , where  $h$  is some computable function of  $k$ .

### 3.1 Properties of the construction

In this section, we prove the important properties of the construction defined above; these properties will form the basis of the proofs of our complexity results in Section 4. We begin by giving a necessary and sufficient condition for a colourful subgraph in  $G_H$  to be isomorphic to  $H$ .

**Lemma 3.1.** *Let  $Y$  be a colourful subset of  $V(G_H)$ . Then  $G_H[Y]$  is isomorphic to a subgraph of  $H$ . Moreover,  $G_H[Y]$  is isomorphic to  $H$  if and only if, for every  $u \in Y \cap H_{(i,\{j,l\})}^{(v,e)}$  and  $w \in Y \cap H_{(i',\{j',l'\})}^{(v',e')}$  such that  $\omega^{-1}f(u)\omega^{-1}f(w) \in E(H)$ , the following conditions are satisfied:*

1. *if  $i = i'$  then  $v = v'$ , and if  $i < i'$  then  $v \prec v'$ , and*
2. *if  $\{j, l\} = \{j', l'\}$  then  $e = e'$ .*

*Proof.* Let  $\pi : Y \rightarrow V(H)$  be the mapping given by  $\pi(u) = \omega^{-1}f(u)$ . We claim first that  $Y$  defines an isomorphism from  $G_H[Y]$  to a subgraph of  $H$ . To verify this claim, it suffices to check that, whenever  $u, w \in Y$  with  $uw \in E(G_H)$ , we also have  $\pi(u)\pi(w) \in E(H)$ . However, by the first condition in the definition of  $E(G_H)$ , we can only have  $uw \in E(G_H)$  if  $\omega^{-1}f(u)\omega^{-1}f(w) \in E(H)$ , or in other words if  $\pi(u)\pi(w) \in E(H)$ , as required.

Now, suppose that  $Y$  is such that, for every  $u \in Y \cap H_{(i,\{j,l\})}^{(v,e)}$  and  $w \in Y \cap H_{(i',\{j',l'\})}^{(v',e')}$  with  $\omega^{-1}f(u)\omega^{-1}f(w) \in E(H)$ , the following conditions are satisfied:

1. if  $i = i'$  then  $v = v'$ , and if  $i < i'$  then  $v \prec v'$ , and
2. if  $\{j, l\} = \{j', l'\}$  then  $e = e'$ .

We claim that in this case  $\pi$  defines an isomorphism from  $G_H[Y]$  to  $H$ . This claim holds provided that, for all  $u, w \in Y$  such that  $\pi(u)\pi(w) \in E(H)$  we also have  $uw \in E(G_H)$ . The assumption that  $\pi(u)\pi(w) \in E(H)$  implies that the first condition for  $uw$  to be an edge of  $G_H$  is satisfied, and the condition we are assuming is exactly the same as the second condition for  $uw$  to be an edge; thus it follows immediately that in this situation we will have  $uw \in E(G_H)$  as required.

Conversely, suppose that the colourful subset  $Y \subset V(G_H)$  induces a subgraph isomorphic to  $H$ . We have already established that  $\pi$  defines an isomorphism from  $G_H[Y]$  to a subgraph of  $H$ , and hence in this case  $\pi$  must be an isomorphism from  $G_H[Y]$  to  $H$ . Suppose that  $u \in Y \cap H_{(i,\{j,l\})}^{(v,e)}$ ,  $w \in Y \cap H_{(i',\{j',l'\})}^{(v',e')}$  and  $\omega^{-1}f(u)\omega^{-1}f(w) \in E(H)$ ; by definition of  $\pi$  this implies that  $\pi(u)\pi(w) \in E(H)$ , and so as  $\pi$  is an isomorphism we must have  $uw \in E(G_H)$ . It then follows immediately from the second condition in the definition of  $E(G_H)$  that

1. if  $i = i'$  then  $v = v'$ , and if  $i < i'$  then  $v \prec v'$ , and
2. if  $\{j, l\} = \{j', l'\}$  then  $e = e'$ ,

as required. □

Using this characterisation, we can now relate the number of colourful subgraphs in  $G_H$  that are isomorphic to  $H$  to the number of cliques in  $G$ .

**Lemma 3.2.** *Let  $G$ ,  $H$ ,  $G_H$  and  $f$  be as described above. Then*

$$\#\text{ColSubInd}(H, G_H, f) = \#\text{Clique}_k(G).$$

*Proof.* We will prove this result by showing that there is a one-to-one correspondence between colourful copies of  $H$  in  $G_H$  with colouring  $f$  and  $k$ -vertex subsets of  $V(G)$  that induce cliques, implying the result. We begin by giving an injective map from the set of  $k$ -vertex subsets inducing cliques in  $G$  to the colourful copies of  $H$  in  $G_H$  with colouring  $f$ ; we will then proceed to give an injective map in the opposite direction.

Given a subset  $X \in V(G)^{(k)}$  that induces a clique in  $G$ , we denote by  $x_1, \dots, x_k$  the elements of  $X$ , with  $x_1 \prec \dots \prec x_k$ . We then claim that the mapping  $\pi$ , where

$$\pi(X) = V'_H \cup \bigcup_{\substack{i,j,l \in [k] \\ j \neq l}} V(H_{(i,\{j,l\})}^{(x_i, x_j x_l)})$$

is an injective mapping from  $k$ -vertex subsets of  $V(G)$  that induce cliques in  $G$  to colourful copies of  $H$  in  $G_H$  (with respect to the colouring  $f$ ).

Observe first that this mapping is well-defined: if  $\{x_1, \dots, x_k\}$  induces a clique in  $G$ , all edges  $x_j x_l$  (with  $j, l \in [k]$ ,  $j \neq l$ ) are edges of  $G$ , and moreover  $x_i$  is incident with  $x_j x_l$  whenever  $i \in \{j, l\}$ , so the specified subgraphs  $H_{(i,\{j,l\})}^{(x_i, x_j x_l)}$  do exist in  $G_H$ .

It is immediate from the definition of  $\pi$  that, for  $X \neq X'$ , we will have  $\pi(X) \neq \pi(X')$ . Thus, to show that  $\pi$  defines an injective map from the set of  $k$ -tuples inducing cliques in  $G$  to the set of colourful copies of  $H$  in  $G_H$ , it remains only to demonstrate that, whenever  $X$  induces a clique in  $G$ ,  $\pi(X)$  must induce a colourful copy of  $H$  in  $G_H$  (with respect to  $f$ ).

We verify first that  $\pi(X)$  is indeed colourful with respect to  $f$ . First note that all vertices of  $V'_H$  have distinct colours, and do not share colours with any vertex in  $V(G_H) \setminus V'_H$ . Next observe that, for each  $(i, \{j, l\}) \in [k] \times [k]^{(2)}$ , there exists a unique pair  $(v, e) \in V(G) \times E(G)$  such that  $\pi(X) \cap V_{(i,\{j,l\})} = V(H_{(i,\{j,l\})}^{(v,e)})$ . Thus,  $\pi(X)$  contains exactly one vertex  $u^{(v,e)}$  (for some  $(v, e) \in V(G) \times E(G)$ ) for each  $u \in V(H) \setminus V'_H$ , and so by definition of  $f$  we see that  $\pi(X)$  is indeed colourful with respect to  $f$ .



Hence, by Lemma 3.1, it suffices to verify that, for every pair of vertices  $u, w \in \pi(X) \setminus V'_H$  such that  $u \in H_{(i, \{j, l\})}^{(v, e)}$ ,  $w \in H_{(i', \{j', l'\})}^{(v', e')}$  and  $\omega^{-1}f(u)\omega^{-1}f(w) \in E(H)$ , the following conditions are satisfied:

1. if  $i = i'$  then  $v = v'$ , and if  $i < i'$  then  $v \prec v'$ , and
2. if  $\{j, l\} = \{j', l'\}$  then  $e = e'$ .

However, these two conditions follow immediately from the definition of  $\pi$ , so  $\pi(X)$  does indeed induce a colourful copy of  $H$ , as required.

We now proceed to define a mapping in the opposite direction, that is an injective mapping from colourful copies of  $H$  in  $G_H$  (with colouring  $f_H$ ) to  $k$ -vertex subsets of  $V(G)$  that induce cliques in  $G$ . Suppose that the colourful subset  $Y \subset V(G_H)$  induces a copy of  $H$ . In order to define our mapping, we prove a series of claims.

**Claim 1.** *For each  $(i, \{j, l\}) \in [k] \times [k]^{(2)}$ , there is a unique pair  $(v, e) \in V(G) \times E(G)$  such that  $Y \cap V_{(i, \{j, l\})} = V(H_{(i, \{j, l\})}^{(v, e)})$ .*

It follows from Lemma 3.1 and the connectedness of  $H[m(u)]$  for each  $u \in V(H)$  that, for each  $(i, \{j, l\}) \in [k] \times [k]^{(2)}$ , there exists a unique pair  $(v, e) \in V(G) \times E(G)$  such that  $Y \cap V_{(i, \{j, l\})} \subseteq V(H_{(i, \{j, l\})}^{(v, e)})$ . Thus, as  $Y$  is colourful, we must in fact have  $Y \cap V_{(i, \{j, l\})} = V(H_{(i, \{j, l\})}^{(v, e)})$  for this  $(v, e)$ , as required.

Hence we can define a mapping  $\sigma_Y : [k] \times [k]^{(2)} \rightarrow V(G) \times E(G)$  by setting  $\sigma_Y(i, \{j, l\})$  to be this unique pair  $(v, e)$  such that  $Y \cap V_{(i, \{j, l\})} = V(H_{(i, \{j, l\})}^{(v, e)})$ ; we denote by  $\sigma_Y^1(i, \{j, l\})$  and  $\sigma_Y^2(i, \{j, l\})$  respectively the first and second elements of this pair.

**Claim 2.** *For each  $i \in [k]$  and every  $\{j, l\}, \{j', l'\} \in [k]^{(2)}$ , we have  $\sigma_Y^1((i, \{j, l\})) = \sigma_Y^1((i, \{j', l'\}))$ . Similarly, for each  $\{j, l\} \in [k]^{(2)}$  and every  $i, i' \in [k]$ , we have  $\sigma_Y^2((i, \{j, l\})) = \sigma_Y^2((i', \{j, l\}))$ .*

To prove this claim we suppose, for a contradiction, there exist  $\{j, l\}$  and  $\{j', l'\}$  such that  $\sigma_Y^1((i, \{j, l\})) \neq \sigma_Y^1((i, \{j', l'\}))$ ; we may assume without loss of generality that  $a_{(i, \{j, l\})}$  and  $a_{(i, \{j', l'\})}$  are adjacent in the grid  $A$ . Now, by definition of  $m$ , there exist vertices  $z, z' \in V(H)$  such that  $z \in m(a_{(i, \{j, l\})})$ ,  $z' \in m(a_{(i, \{j', l'\})})$  and  $zz' \in E(H)$ . Let  $y$  and  $y'$  be the vertices of  $Y$  whose colours match  $z$  and  $z'$  respectively, that is  $y = f|_Y^{-1}\omega(z)$  and  $y' = f|_Y^{-1}\omega(z')$  (as  $Y$  is colourful, the restriction of  $f$  to  $Y$  is a bijection). Note that we

must have  $y \in H_{(i,\{j,l\})}^{\sigma_Y((i,\{j,l\}))}$  and  $y' \in H_{(i,\{j',l'\})}^{\sigma_Y((i,\{j',l'\}))}$ . By our assumption,  $\sigma_Y^1((i,\{j,l\})) \neq \sigma_Y^1((i,\{j',l'\}))$ , but by Lemma 3.1 this then means that  $Y$  cannot induce a copy of  $H$  in  $G_H$ , giving a contradiction. Thus we know that, for each  $i \in [k]$ , we must have  $\sigma_Y^1((i,\{j,l\})) = \sigma_Y^1((i,\{j',l'\}))$  for all  $\{j,l\}, \{j',l'\} \in [k]^{(2)}$ . An analogous argument can be used to show that, for each  $\{j,l\} \in [k]^{(2)}$ , we must have  $\sigma_Y^2((i,\{j,l\})) = \sigma_Y^2((i',\{j,l\}))$  for every  $i, i' \in [k]$ .

Hence we can define the mapping  $\tau_Y^1 : [k] \rightarrow V(G)$  by setting  $\tau_Y^1(i)$  to be the unique vertex  $v \in V(G)$  such that, for any value of  $\{j,l\} \in [k]^{(2)}$ ,  $\sigma_Y^1((i,\{j,l\})) = v$ , and the mapping  $\tau_Y^2 : [k]^{(2)} \rightarrow E(G)$  by setting  $\tau_Y^2(\{j,l\})$  to be the unique edge  $e \in E(G)$  such that, for any value of  $i \in [k]$ ,  $\sigma_Y^2((i,\{j,l\})) = e$ .

**Claim 3.** *For  $i \neq i'$ , we have  $\tau_Y^1(i) \neq \tau_Y^1(i')$ .*

We argue that, for  $1 \leq i \leq k-1$ , we must have  $\tau_Y^1(i) \prec \tau_Y^1(i+1)$ , implying that  $\tau_Y^1(1) \prec \tau_Y^1(2) \prec \dots \prec \tau_Y^1(k)$  and in particular that no two of these vertices are the same. As in the proof of the preceding claim, we observe that by definition of  $m$ , for each  $i \in \{1, \dots, k-1\}$ , there exist  $z, z' \in V(H)$  such that  $z \in m(a_{(i,\{1,2\})})$ ,  $z' \in m(a_{(i+1,\{1,2\})})$  and  $zz' \in E(H)$ ; as before, we set  $y = f|_{\bar{Y}}^{-1}\omega(z)$  and  $y' = f|_{\bar{Y}}^{-1}\omega(z')$ . Note that we have  $y \in H_{(i,\{1,2\})}^{(\tau_Y^1(i), \tau_Y^2(\{1,2\}))}$  and  $y' \in H_{(i+1,\{1,2\})}^{(\tau_Y^1(i+1), \tau_Y^2(\{1,2\}))}$ . Since  $i < i+1$ , it now follows from Lemma 3.1 that  $\tau_Y^1(i) \prec \tau_Y^1(i+1)$ , as required.

**Claim 4.**  $V_Y = \{\tau_Y^1(i) : 1 \leq i \leq k\}$  induces a  $k$ -clique in  $G$ .

It suffices to verify that, for each  $1 \leq i < i' \leq k$ ,  $\tau_Y^1(i)\tau_Y^1(i') \in E(G)$ : since  $G$  is a simple graph with no self-loops this will imply that  $\tau_Y^1(i) \neq \tau_Y^1(i')$  and hence we will have  $|V_Y| = k$ .

Suppose, therefore, that  $1 \leq i < i' \leq k$ , and consider the edge  $e \in E(G)$  such that  $\tau_Y^2(\{i, i'\}) = e$ . We claim that  $e$  must in fact be  $\tau_Y^1(i)\tau_Y^1(i')$ . To see that this is true, note that, by definition, we have  $\sigma_Y((i, \{i, i'\})) = (\tau_Y^1(i), e)$  and  $\sigma_Y((i', \{i, i'\})) = (\tau_Y^1(i'), e)$ ; this implies that the subgraphs  $H_{(i,\{i,i'\})}^{(\tau_Y^1(i), e)}$  and  $H_{(i',\{i,i'\})}^{(\tau_Y^1(i'), e)}$  must both belong to  $G_H$ . However, by definition of  $V(G_H)$ , the fact that  $i \in \{i, i'\}$  and  $i' \in \{i, i'\}$  implies that both  $\tau_Y^1(i)$  and  $\tau_Y^1(i')$  must be incident with  $e$ , and since  $\tau_Y^1(i) \neq \tau_Y^1(i')$  this is only possible if  $\tau_Y^1(i)\tau_Y^1(i') \in E(G)$ , as required, implying that  $V_Y$  does indeed induce a  $k$ -

clique in  $G$ . Note that this argument further implies that, for all  $j, l \in [k]$ ,  $\tau_Y^2(\{j, l\}) = \tau_Y^1(j)\tau_Y^1(l)$ .

**Claim 5.** *The mapping  $Y \mapsto (\tau_Y^1(1), \dots, \tau_Y^1(k))$  is injective.*

It follows from the preceding claims that this mapping is a well-defined mapping from colourful subsets of  $V(G_H)$  such that  $G_H[Y] \cong H$  to  $k$ -tuples of  $G$  that induce cliques, so the proof of this claim will complete the proof of the result.

To see that this claim holds, suppose that  $Y$  and  $Y'$  are distinct colourful subsets of  $G_H$  with respect to the colouring  $f$ , such that  $G_H[Y], G_H[Y'] \cong H$ . By the reasoning above, we know that there are functions  $\tau_Y^1, \tau_{Y'}^1 : [k] \rightarrow V(G)$  such that, for each  $(i, \{j, l\}) \in [k] \times [k]^{(2)}$ ,

$$Y \cap V_{(i, \{j, l\})} = V(H_{(i, \{j, l\})}^{(\tau_Y^1(i), \tau_Y^1(j), \tau_Y^1(l))})$$

and

$$Y' \cap V_{(i, \{j, l\})} = V(H_{(i, \{j, l\})}^{(\tau_{Y'}^1(i), \tau_{Y'}^1(j), \tau_{Y'}^1(l))}).$$

Since

$$\begin{aligned} Y &= V'_H \cup \bigcup_{(i, \{j, l\}) \in [k] \times [k]^{(2)}} (Y \cap V_{(i, \{j, l\})}) \\ &= V'_H \cup \bigcup_{(i, \{j, l\}) \in [k] \times [k]^{(2)}} V(H_{(i, \{j, l\})}^{(\tau_Y^1(i), \tau_Y^1(j), \tau_Y^1(l))}) \end{aligned}$$

and

$$\begin{aligned} Y' &= V'_H \cup \bigcup_{(i, \{j, l\}) \in [k] \times [k]^{(2)}} (Y' \cap V_{(i, \{j, l\})}) \\ &= V'_H \cup \bigcup_{(i, \{j, l\}) \in [k] \times [k]^{(2)}} V(H_{(i, \{j, l\})}^{(\tau_{Y'}^1(i), \tau_{Y'}^1(j), \tau_{Y'}^1(l))}), \end{aligned}$$

the fact that  $Y$  and  $Y'$  are not equal implies that  $\tau_Y^1 \neq \tau_{Y'}^1$ . It therefore follows immediately that the images of  $Y$  and  $Y'$  under our mapping are distinct.  $\square$

We now prove a simple fact about the number of colourful copies of graphs from some collection  $\mathcal{H}$  in  $G_H$ , provided that  $\mathcal{H}$  satisfies certain conditions.

**Lemma 3.3.** *Let  $\mathcal{H}$  be a collection of graphs on  $k$  vertices such that  $H \in \mathcal{H}$  and there is no  $H' \in \mathcal{H}$  such that  $H'$  is a proper subgraph of  $H$ . Then*

$$\#\text{ColStrEmb}(\mathcal{H}, G_H, f) = \#\text{ColStrEmb}(\mathcal{H}^H, G_H, f).$$

*Proof.* We know from Lemma 3.1 that any colourful subset in  $G_H$  induces a subgraph of  $H$ ; thus, by our assumption that  $\mathcal{H}$  does not contain any proper subgraphs of  $H$ , it follows immediately that the number of colourful labelled copies of graphs from  $\mathcal{H}$  in  $G_H$  (with respect to the colouring  $f$ ) is exactly equal to the number of colourful labelled copies of graphs from  $\mathcal{H}^H$  in  $G_H$  (with colouring  $f$ ).  $\square$

## 4 Complexity results

In this section, we make use of the construction described in Section 3 above to prove a number of complexity results. We will give a treewidth-based condition that guarantees hardness of exact counting for problems falling within our model, and dichotomies for the complexity of decision and approximate counting in the multicolour case, which have implications for the corresponding uncoloured problems. We begin, however, with some auxiliary results.

### 4.1 Auxiliary results

Here we give two auxiliary results which will be used to prove the complexity results below. First, we will make use of the following lemma, proved in [14].

**Lemma 4.1.** *Let  $\mathcal{H}$  be a collection of labelled graphs, and  $(H, \pi) \in \mathcal{H}$  a labelled  $k$ -vertex graph. Set*

$$\alpha_H = |\{\sigma : \sigma \text{ a permutation on } [k], \exists (H, \pi') \in \mathcal{H}^H \text{ such that } \pi' \circ \sigma^{-1} \circ \pi^{-1} \text{ defines an automorphism on } H\}|.$$

*Then, for any graph  $G$ ,*

$$\#\text{StrEmb}(\mathcal{H}^H, G) = \alpha_H \cdot \#\text{SubInd}(H, G),$$

Moreover, if  $G$  is equipped with a  $k$ -colouring  $f$ , then

$$\#\text{ColStrEmb}(\mathcal{H}^H, G, f) = \alpha_H \cdot \#\text{ColSubInd}(H, G, f).$$

Note that the value of  $\alpha_H$ , as defined in the statement of this lemma, can be computed from  $H$  and  $\mathcal{H}$  in time bounded only by some computable function of  $k$ .

The following easy proposition will also be used in several of our proofs below.

**Proposition 4.2.** *Let  $\Phi$  be a family  $(\phi_1, \phi_2, \dots)$  of functions  $\phi_k : \{0, 1\}^{\binom{k}{2}} \rightarrow \{0, 1\}$ , such that the function mapping  $k \mapsto \phi_k$  is computable. Suppose that there exists a computable function  $g$  such that, for every  $t \in \mathbb{N}$ , there exists  $l \leq g(t)$  such that some minimal unlabelled subgraph satisfying  $\phi_l$  has treewidth greater than  $t$ . Then, for every  $j \in \mathbb{N}$  there exists  $k \in \mathbb{N}$  such that some minimal unlabelled subgraph satisfying  $\phi_k$  contains the  $(j \times \binom{j}{2})$  grid as a minor; moreover, the value of this  $k$  is bounded by some computable function of  $j$ .*

*Proof.* By the Excluded Grid Theorem (Theorem 2.11), there exists a computable function  $w$  such that every graph with treewidth at least  $w(\binom{j}{2})$  contains as a minor the  $(\binom{j}{2} \times \binom{j}{2})$  grid, and hence the  $(j \times \binom{j}{2})$  grid. Thus, by our assumptions on  $\Phi$ , there exists  $k \leq g(w(\binom{j}{2}))$  such that some minimal unlabelled graph  $(H, \pi)$ , satisfying  $\phi_k$ , contains the  $(j \times \binom{j}{2})$  grid as a minor.  $\square$

## 4.2 Hardness of exact counting

In this section, we give a sufficient condition for subgraph counting problems in our model to be  $\#\text{W}[1]$ -complete: if there is no  $t \in \mathbb{N}$  such that, for every  $k \in \mathbb{N}$ , all minimal unlabelled subgraphs satisfying  $\phi_k$  have treewidth at most  $t$ , then  $p\text{-}\#\text{INDUCED SUBGRAPH WITH PROPERTY}(\Phi)$  is  $\#\text{W}[1]$ -complete, and the same is true for  $p\text{-}\#\text{MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY}(\Phi)$ .

**Theorem 4.3.** *Let  $\Phi$  be a family  $(\phi_1, \phi_2, \dots)$  of functions  $\phi_k : \{0, 1\}^{\binom{k}{2}} \rightarrow \{0, 1\}$ , such that the function mapping  $k \mapsto \phi_k$  is computable. Suppose that*

there exists a computable function  $g$  such that, for every  $t \in \mathbb{N}$ , there exists  $k' \leq g(t)$  such that some minimal unlabelled subgraph satisfying  $\phi_{k'}$  has treewidth greater than  $t$ . Then  $p$ -#INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) and  $p$ -#MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) are #W[1]-complete.

*Proof.* We prove the result by means of an fpt-Turing reduction from the #W[1]-complete problem  $p$ -#CLIQUE. Note that, by Lemma 2.4, it suffices to give a reduction from  $p$ -#CLIQUE to  $p$ -#MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ).

Let the graph  $G$  be the input to an instance of  $p$ -#CLIQUE with parameter  $k$ . By Proposition 4.2, we know that there is  $k'$  (bounded by some computable function of  $k$ ) such that some minimal unlabelled subgraph satisfying  $\phi_{k'}$ , which we shall call  $H$ , contains the  $(k \times \binom{k}{2})$  grid as a minor and, moreover, that there is no  $H'$  satisfying  $\phi_{k'}$  such that  $H'$  is a proper subgraph of  $H$ . Let  $\mathcal{H}$  be the set of all  $k'$ -vertex labelled graphs satisfying  $\phi_{k'}$ .

Recall that the graph  $G_H$  and colouring  $f$  can be constructed in time  $h(k) \cdot n^{O(1)}$ , where  $h$  is some computable function. Given the graph  $G_H$ , the colouring  $f$  and an oracle to  $p$ -#MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ), it is now straightforward to compute the number of  $k$ -cliques in  $G$ , since

$$\begin{aligned}
\#Clique_k(G, f) &= \#ColSubInd(H, G_H, f) \\
&\quad \text{by Lemma 3.2} \\
&= \frac{1}{\alpha_H} \#ColStrEmb(\mathcal{H}^H, G_H, f) \\
&\quad \text{by Lemma 4.1} \\
&= \frac{1}{\alpha_H} \#ColStrEmb(\mathcal{H}, G_H, f) \\
&\quad \text{by Lemma 3.3,}
\end{aligned}$$

where

$$\alpha_H = |\{\sigma : \sigma \text{ a permutation on } [k], \exists (H, \pi') \in \mathcal{H}^H \text{ such that } \pi' \circ \sigma^{-1} \circ \pi^{-1} \text{ defines an automorphism on } H\}|.$$

Recall that  $\alpha_H$  can be computed from  $H$  and  $\mathcal{H}$  in time depending only on a computable function of  $k'$ . Note also that the value of the parameter in the oracle call used to evaluate  $\#ColStrEmb(\mathcal{H}, G_H, f)$  is at most

$g(w(\binom{k}{2}))$ , which is a computable function of  $k$  only. This therefore gives an fpt-Turing reduction from  $p\text{-}\#\text{CLIQUE}$  to  $p\text{-}\#\text{MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY}(\Phi)$ , as required.  $\square$

### 4.3 A dichotomy for colourful decision when $\Phi$ is monotone

In this section we turn to the complexity of the decision versions of problems in our model, and give a dichotomy for the complexity of such problems in the multicolour case when the property in question is monotone. For the definition of fpt-reductions for parameterised decision problems, we again refer the reader to [11].

**Theorem 4.4.** *Let  $\Phi$  be a family  $(\phi_1, \phi_2, \dots)$  of functions  $\phi_k : \{0, 1\}^{\binom{k}{2}} \rightarrow \{0, 1\}$ , such that the function mapping  $k \mapsto \phi_k$  is computable, and suppose further that  $\Phi$  is a monotone property. Then  $p\text{-MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY}(\Phi)$  is in FPT if there exists an integer  $t$  such that, for each  $k \in \mathbb{N}$ , all minimal unlabelled graphs satisfying  $\phi_k$  have treewidth at most  $t$ ; otherwise  $p\text{-MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY}(\Phi)$  is W[1]-complete.*

*Proof.* The first part of the result is precisely Corollary 2.14. Thus it remains to show that if this condition is not satisfied, we must instead have that  $p\text{-MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY}(\Phi)$  is W[1]-complete. We prove W[1]-completeness by means of an fpt-reduction from  $p\text{-CLIQUE}$ . Let  $G$  be the input in an instance of  $p\text{-CLIQUE}$  with parameter  $k$ . By assumption, there is no integer  $t$  such that, for each  $k' \in \mathbb{N}$ , all minimal unlabelled graphs satisfying  $\phi_{k'}$  have treewidth at most  $t$ . It then follows from Proposition 4.2 that there exists  $k' \in \mathbb{N}$  (where  $k'$  is at most some computable function of  $k$ ) such that some minimal unlabelled subgraph  $(H, \pi)$  satisfying  $\phi_{k'}$  contains the  $(k \times \binom{k}{2})$  grid as a minor.

Now, we can construct the graph  $G_H$  and colouring  $f$  in time  $h(k) \cdot n^{O(1)}$ , where  $h$  is some computable function. Moreover, we know from Lemma 3.2 that

$$\#\text{ColSubInd}(H, G_H, f) = \#\text{Clique}_k(G),$$

so in particular the number of colourful subgraphs of  $G_H$  (with respect to the colouring  $f$ ) that are isomorphic to  $H$  is non-zero if and only if the number of  $k$ -cliques in  $G$  is non-zero. Since the parameter,  $k'$ , in the instance of

$p$ -MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) is bounded by a computable function of  $k$ , this gives a fpt-reduction from  $p$ -CLIQUE to  $p$ -MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) in this second case, as required. This completes the proof of the theorem.  $\square$

#### 4.4 Approximability of monotone properties

In this section we consider the conditions under which there exists an FPTRAS for  $p$ -#INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ), under the assumption that  $\Phi$  is a nice, monotone property. We begin by building on the results of Section 4.3 to show that the existence of an efficient algorithm to solve  $p$ -MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) in this case implies the existence of an FPTRAS for  $p$ -#MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ). Combined with Corollary 2.3, this gives a dichotomy for the approximability of multicolour subgraph counting problems defined by a monotone property, subject to the assumption that  $W[1] \not\subseteq$  randomised FPT.

**Theorem 4.5.** *Assume that  $FPT \neq W[1]$ . Let  $\Phi$  be a family  $(\phi_1, \phi_2, \dots)$  of functions  $\phi_k : \{0, 1\}^{\binom{k}{2}} \rightarrow \{0, 1\}$ , such that the function mapping  $k \mapsto \phi_k$  is computable, and suppose further that  $\Phi$  is a nice, monotone property. Then, if  $p$ -MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) is in FPT, there is an FPTRAS for  $p$ -#MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ).*

*Proof.* By Theorem 4.4, we know that  $p$ -MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) is in FPT if and only if there exists an integer  $t$  such that, for each  $k \in \mathbb{N}$ , all minimal unlabelled graphs satisfying  $\phi_k$  have treewidth at most  $t$ ; by our assumption that  $\Phi$  is nice, the same is true for the minimal labelled subgraphs satisfying  $\phi_k$ . Thus it remains only to show that there exists an FPTRAS for  $p$ -#MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) if  $\Phi$  satisfies this condition; this is exactly the statement of Theorem 2.15.  $\square$

We now apply the previous result to give a sufficient condition for the existence of an FPTRAS for  $p$ -#INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) when  $\Phi$  is a monotone property.

**Corollary 4.6.** *Let  $\Phi$  be a family  $(\phi_1, \phi_2, \dots)$  of functions  $\phi_k : \{0, 1\}^{\binom{k}{2}} \rightarrow \{0, 1\}$ , such that the function mapping  $k \mapsto \phi_k$  is computable, and suppose*



further that  $\Phi$  is a nice, monotone property. Then, if  $p$ -MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) is in FPT, there is an FPTRAS for  $p$ -#INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ).

*Proof.* We know from Theorem 4.5 that, if  $p$ -MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) is in FPT, there is an FPTRAS for  $p$ -#MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ). Thus, by Proposition 2.6, we must in this case also have an FPTRAS for  $p$ -#INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ).  $\square$

## 5 Conclusions and Open Problems

We have proved a range of results concerning the complexity of subgraph counting problems in which the treewidth of minimal subgraphs having the desired property cannot be bounded by any constant. We demonstrated that, for properties  $\Phi$  of this kind, both  $p$ -#INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) and  $p$ -#MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) are #W[1]-complete, and that the multicolour decision problem  $p$ -MULTICOLOUR INDUCED SUBGRAPH WITH PROPERTY( $\Phi$ ) is W[1]-complete.

These results contrast with positive results for decision and approximation of monotone properties for which all minimal subgraphs satisfying the property have bounded treewidth; combining our results with these positive results gives a dichotomy for the decision version of the problem and, with an additional restriction on the relationship between the minimal labelled and unlabelled subgraphs having the property, we obtain another dichotomy for multicolour approximate counting.

There remain two natural open questions to ask concerning the existence of an FPTRAS for such problems. First of all, our dichotomy for the existence of an FPTRAS for multicoloured subgraph counting problems is only valid for monotone properties for which also satisfy our definition of a “nice” property: is the fact that the decision version is in FPT enough to guarantee the existence of an FPTRAS more generally in the multicolour case? Secondly, we gave a sufficient condition for the existence of an FPTRAS for monotone subgraph counting problems in the uncoloured version: is it possible to prove a dichotomy in this case as well?

Our #W[1]-completeness results contribute to the categories of parameterised counting problems that are known to be computationally difficult. It

was asked in [14] whether there are any non-trivial subgraph counting problems in the uncoloured version of this model which admit FPT algorithms, and the results here add another powerful restriction which must be satisfied by any such problem, but the original question remains open.

## Acknowledgements

I am very grateful to Mark Jerrum for his helpful comments on a first draft of this paper.

## References

- [1] Noga Alon and Shai Gutner, *Balanced hashing, color coding and approximate counting*, Parameterized and Exact Computation (Jianer Chen and Fedor V. Fomin, eds.), Lecture Notes in Computer Science, vol. 5917, Springer Berlin Heidelberg, 2009, pp. 1–16.
- [2] Noga Alon, Raphael Yuster, and Uri Zwick, *Color-coding*, Journal of the ACM **42** (1995), no. 4, 844–856.
- [3] V. Arvind and Venkatesh Raman, *Approximation algorithms for some parameterized counting problems*, ISAAC 2002 (P. Bose and P. Morin, eds.), LNCS, vol. 2518, Springer-Verlag Berlin Heidelberg, pp. 453–464.
- [4] Nadja Betzler, Rene van Bevern, Michael R. Fellows, Christian Komusiewicz, and Rolf Niedermeier, *Parameterized algorithmics for finding connected motifs in biological networks*, IEEE/ACM Trans. Comput. Biol. Bioinformatics **8** (2011), no. 5, 1296–1308.
- [5] Yijia Chen, Marc Thurley, and Mark Weyer, *Understanding the complexity of induced subgraph isomorphisms*, Automata, Languages and Programming (Luca Aceto, Ivan Damgrd, Leslie Ann Goldberg, Magnús M. Halldrsson, Anna Inglsdttir, and Igor Walukiewicz, eds.), Lecture Notes in Computer Science, vol. 5125, Springer Berlin Heidelberg, 2008, pp. 587–596.
- [6] Radu Curticapean, *Counting matchings of size  $k$  is  $\#W[1]$ -hard*, Automata, Languages, and Programming (Fedor V. Fomin, Rsi Freivalds,

Marta Kwiatkowska, and David Peleg, eds.), Lecture Notes in Computer Science, vol. 7965, Springer Berlin Heidelberg, 2013, pp. 352–363.

- [7] Víctor Dalmau and Peter Jonsson, *The complexity of counting homomorphisms seen from the other side*, Theoretical Computer Science **329** (2004), no. 13, 315 – 323.
- [8] Rod G. Downey and Michael R. Fellows, *Fixed-parameter tractability and completeness II: On completeness for  $W[1]$* , Theoretical Computer Science **141** (1995), no. 12, 109 – 131.
- [9] M. Fellows, D. Hermelin, F. Rosamond, and S. Vialette, *On the fixed-parameter intractability and tractability of multiple-interval graph properties*, Theoretical Computer Science **410** (2009), 53–61.
- [10] J. Flum and M. Grohe, *The parameterized complexity of counting problems*, SIAM Journal on Computing **33** (2004), no. 4, 892–922.
- [11] ———, *Parameterized complexity theory*, Springer, 2006.
- [12] Martin Grohe, *The complexity of homomorphism and constraint satisfaction problems seen from the other side*, J. ACM **54** (2007), no. 1, 1:1–1:24.
- [13] Mark Jerrum and Kitty Meeks, *The parameterised complexity of counting connected subgraphs and graph motifs*, arXiv.1308.1575v1 [cs.CC], August 2013.
- [14] ———, *Some hard families of parameterised counting problems*, arXiv.1310.6524 [cs.CC], October 2013.
- [15] J. Plehn and B. Voigt, *Finding minimally weighted subgraphs*, Graph-Theoretic Concepts in Computer Science, WG 90 (R. Möhring, ed.), Lecture Notes in Computer Science, vol. 484, Springer-Verlag, 1990, pp. 18–29.
- [16] Neil Robertson and P.D Seymour, *Graph minors. V. excluding a planar graph*, Journal of Combinatorial Theory, Series B **41** (1986), no. 1, 92 – 114.